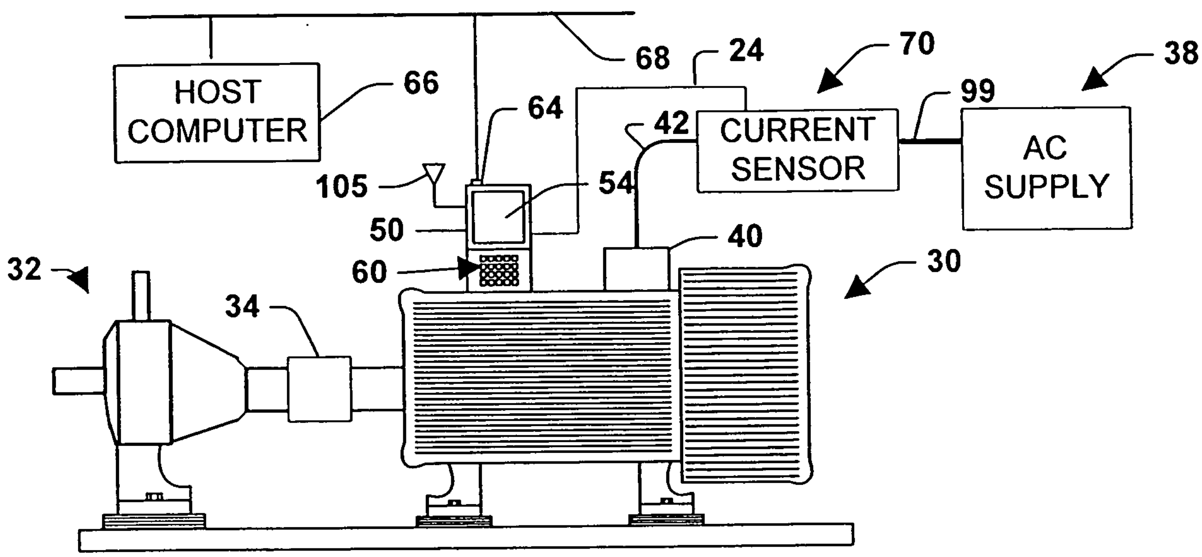

68
24
70
38
HOST
COMPUTER
66
64
42
CURRENT
SENSOR
99
AC
SUPPLY
105
54
50
40
60
30
32
34

Fig. 1a

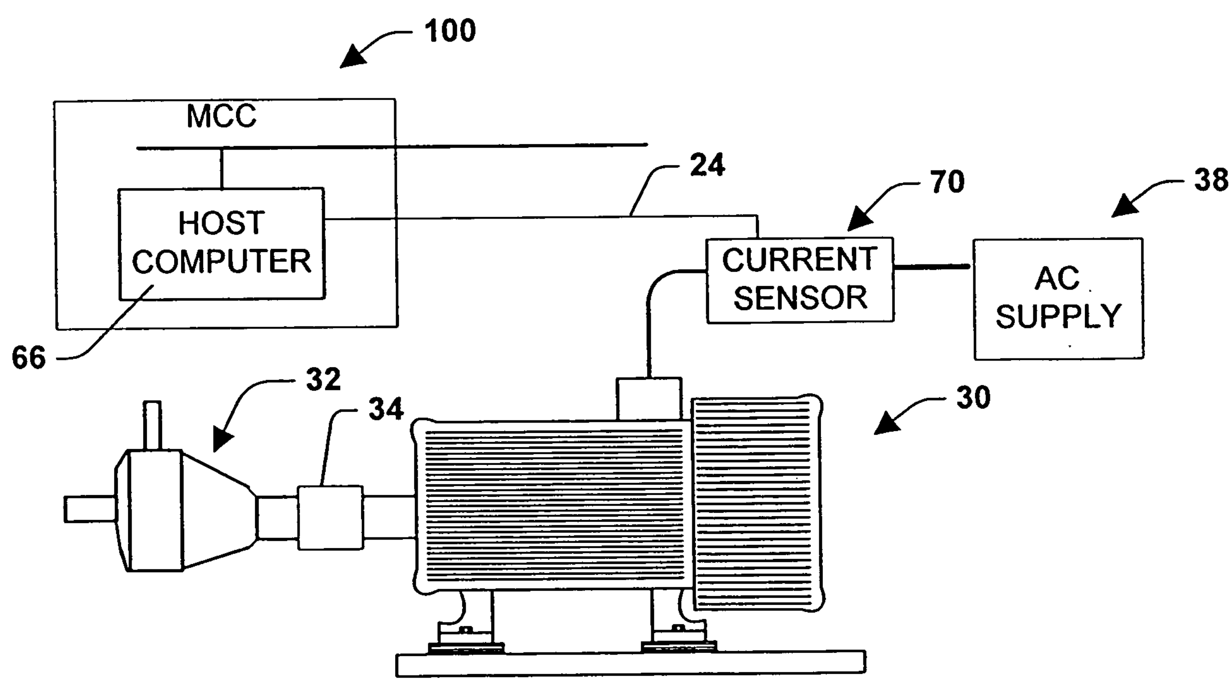

100
MCC
24
70
38
HOST
COMPUTER
CURRENT
SENSOR
AC
SUPPLY
66
32
34
30

Fig. 1b

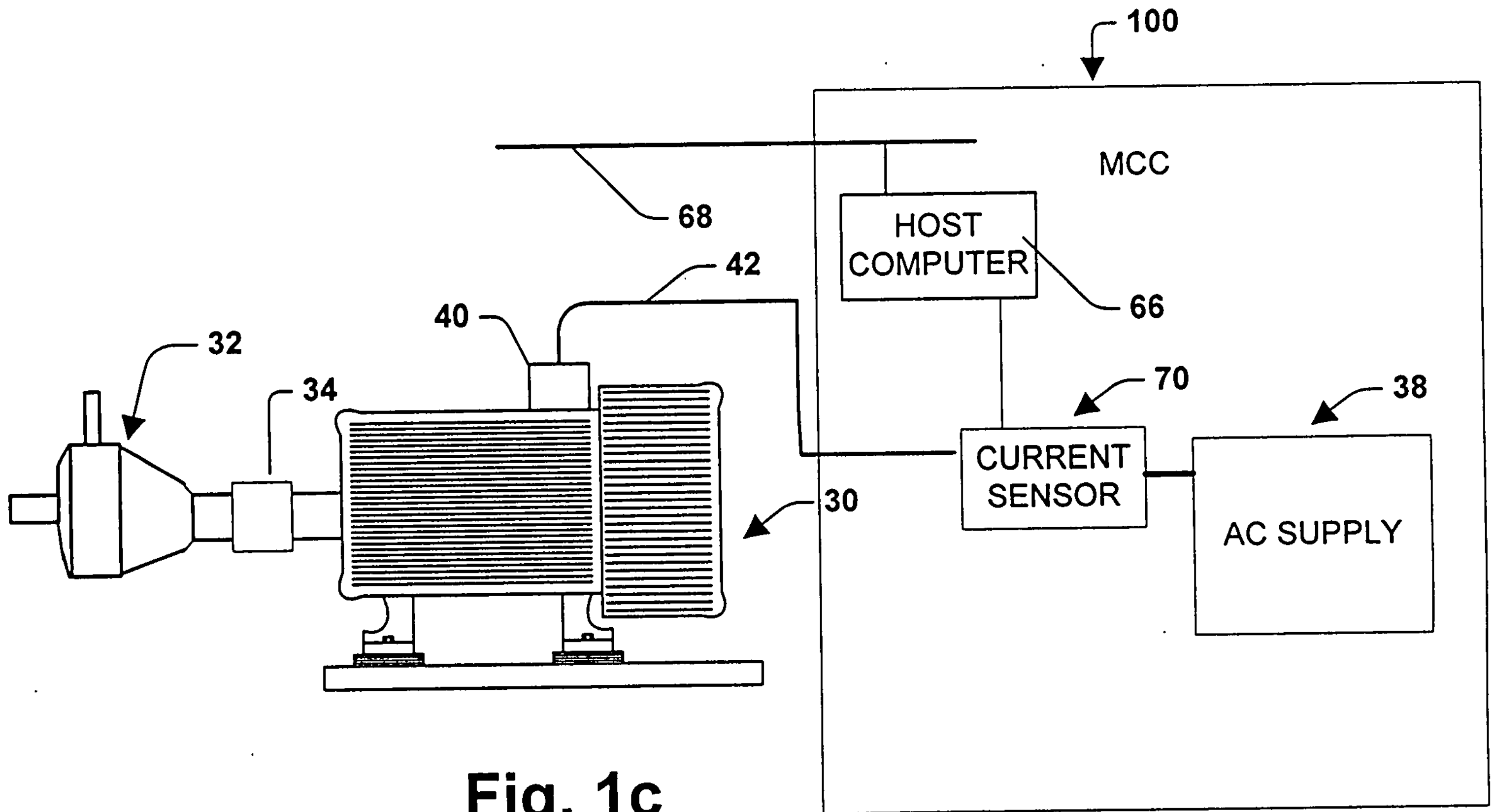

100
MCC
HOST COMPUTER
66
68
42
40
32
34
70
38
CURRENT SENSOR
AC SUPPLY
30

Fig. 1c

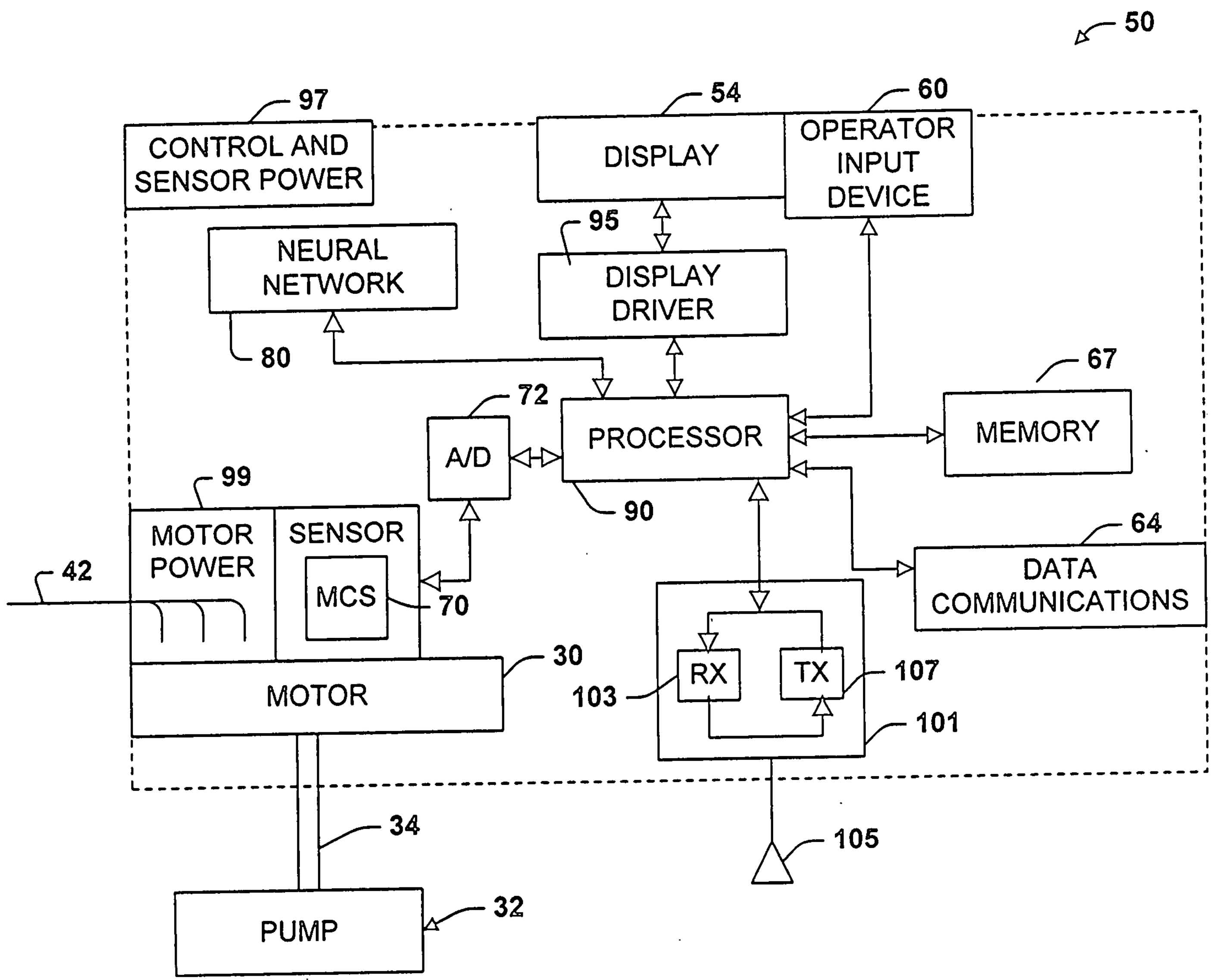

50
97
CONTROL AND SENSOR POWER
54
DISPLAY
60
OPERATOR INPUT DEVICE
95
DISPLAY DRIVER
NEURAL NETWORK
80
67
MEMORY
72
A/D
PROCESSOR
90
99
MOTOR POWER
SENSOR
MCS
70
42
64
DATA COMMUNICATIONS
30
MOTOR
103
RX
TX
107
101
34
105
PUMP
32

Fig. 2

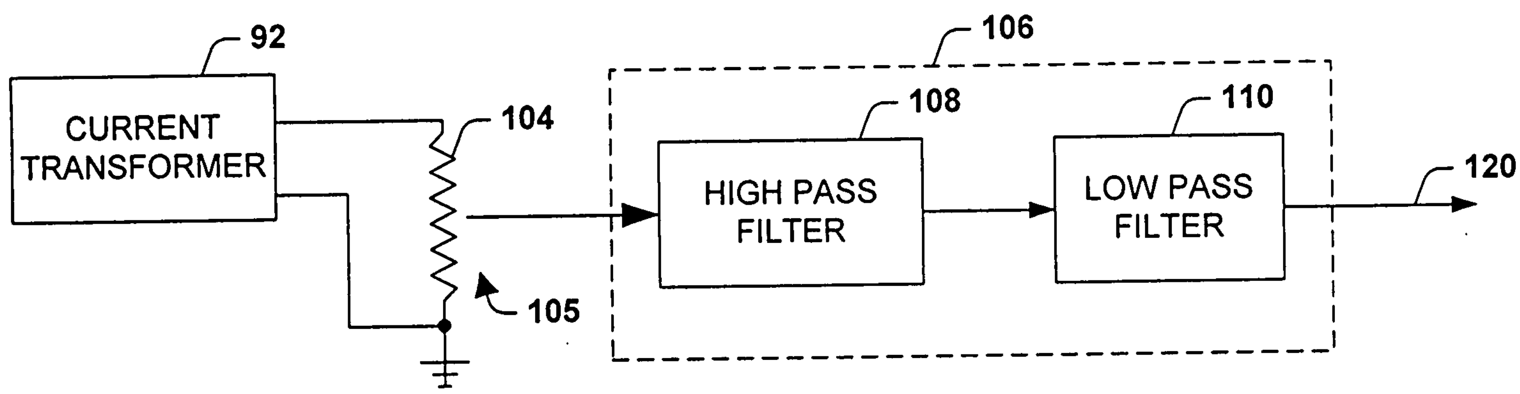

92
CURRENT TRANSFORMER
104
105
106
108
HIGH PASS FILTER
110
LOW PASS FILTER
120

Fig. 3

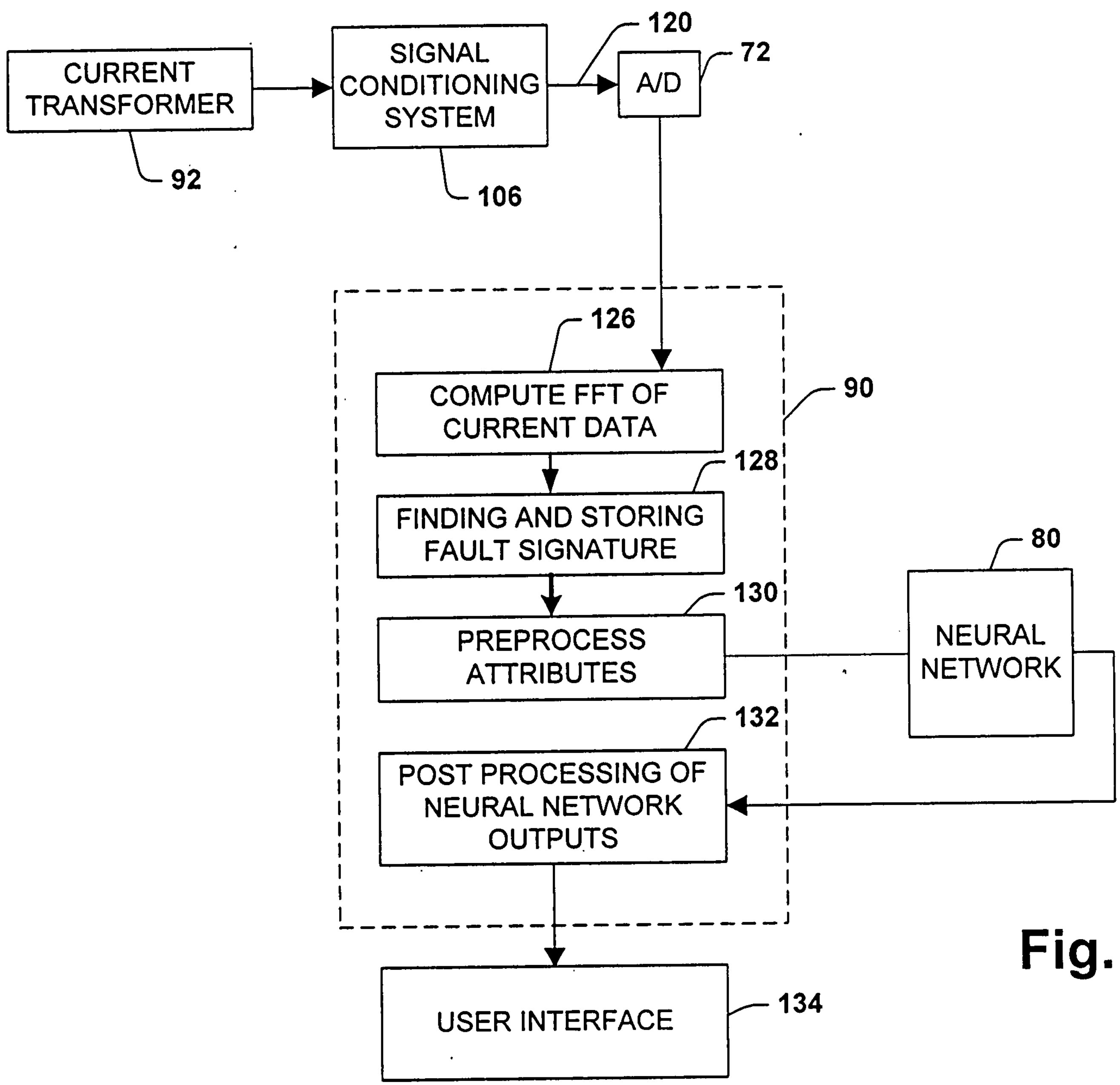

CURRENT TRANSFORMER
92
SIGNAL CONDITIONING SYSTEM
106
120
A/D
72
90
126
COMPUTE FFT OF CURRENT DATA
128
FINDING AND STORING FAULT SIGNATURE
130
PREPROCESS ATTRIBUTES
132
POST PROCESSING OF NEURAL NETWORK OUTPUTS
80
NEURAL NETWORK
USER INTERFACE
134

Fig. 4a

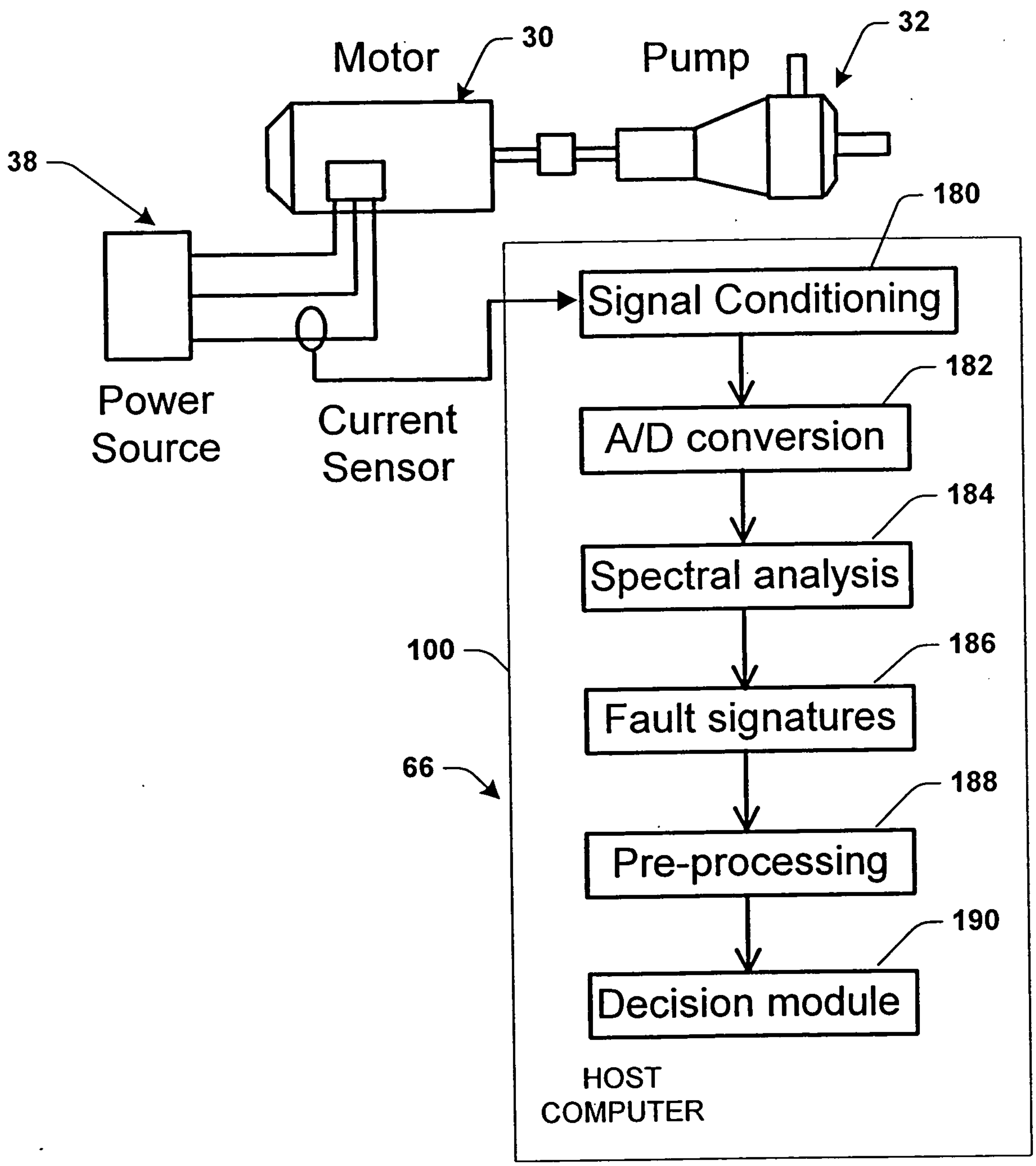
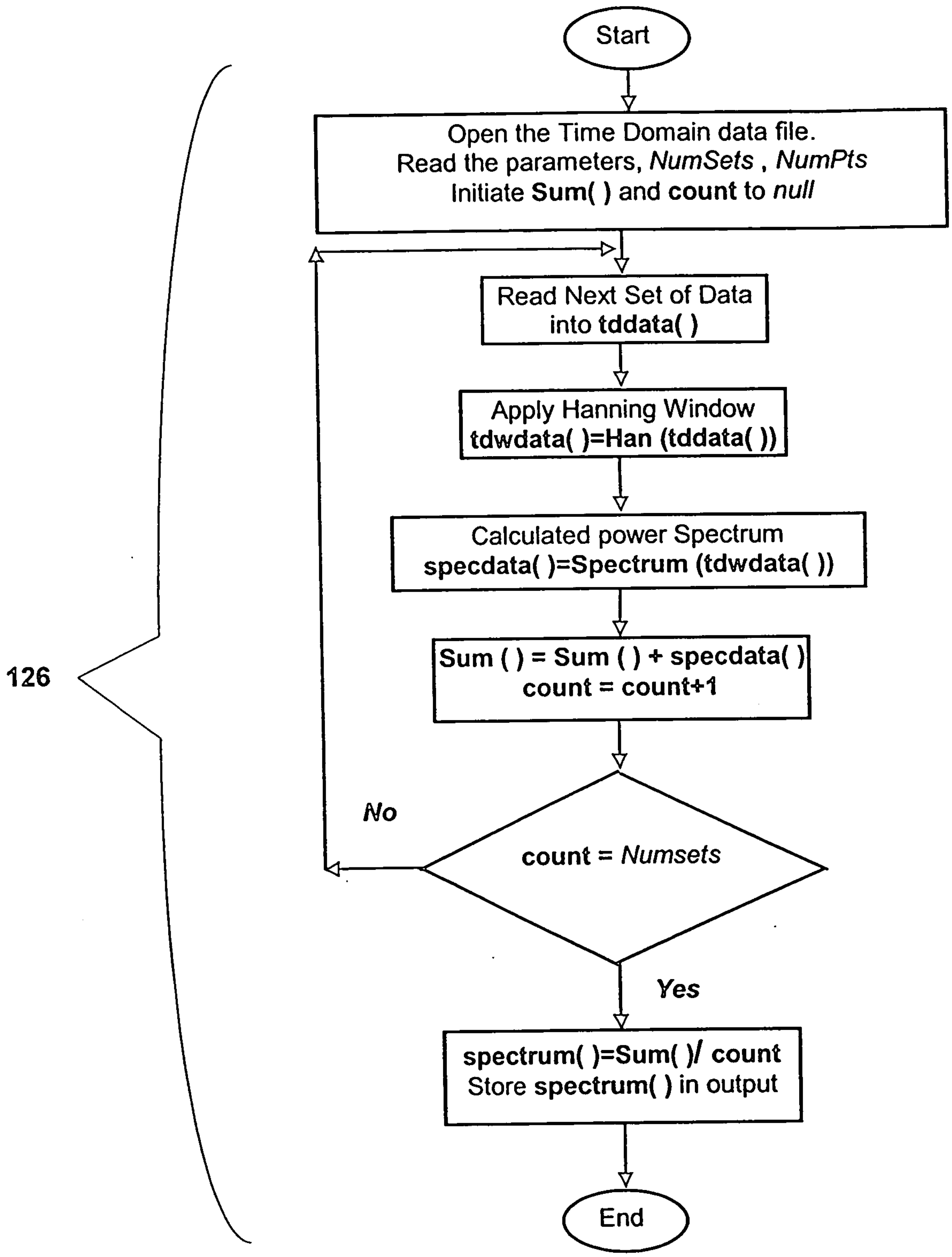

Motor
30
Pump
32
38
180
Signal Conditioning
182
A/D conversion
184
Spectral analysis
186
Fault signatures
188
Pre-processing
190
Decision module
Power
Source
Current
Sensor
100
66
HOST
COMPUTER

Fig. 4b



Start
Open the Time Domain data file.
Read the parameters, NumSets , NumPts
Initiate Sum( ) and count to null
Read Next Set of Data into tddata( )
Apply Hanning Window tdwdata( )=Han (tddata( ))
Calculated power Spectrum specdata( )=Spectrum (tdwdata( ))
Sum ( ) = Sum ( ) + specdata( ) count = count+1
count = Numsets
No
Yes
spectrum( )=Sum( )/ count Store spectrum( ) in output
End
126

Fig. 5

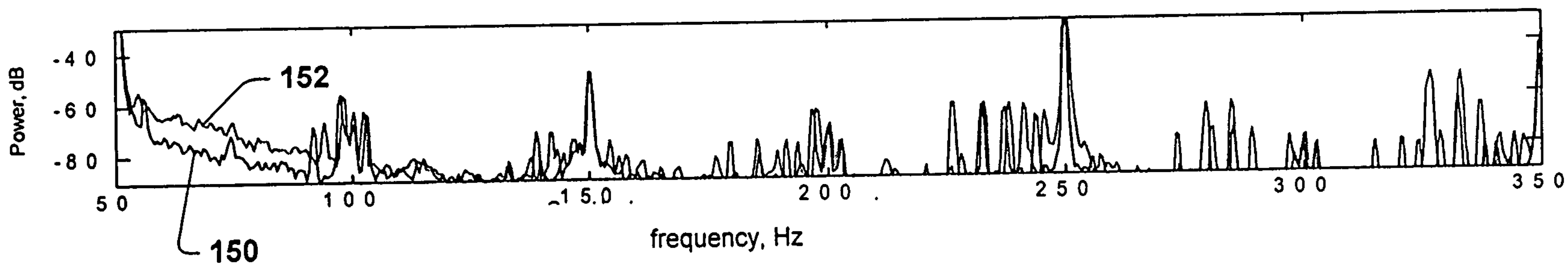

152
150
Power, dB
-40
-60
-80
50
100
150
200
250
300
350
frequency, Hz

Fig. 6a

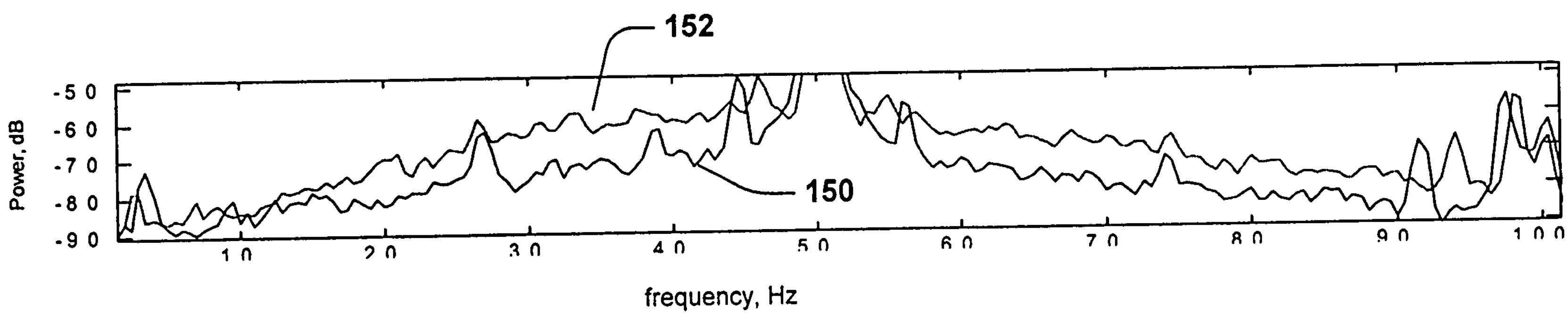
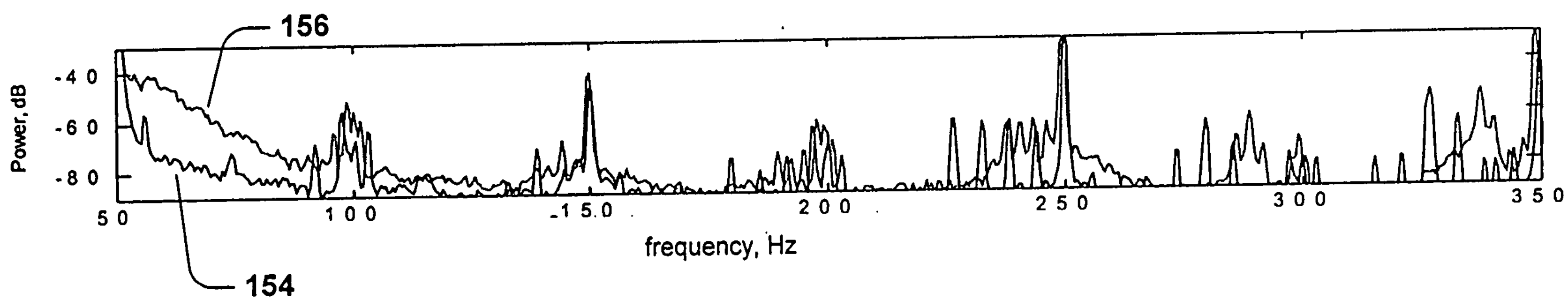
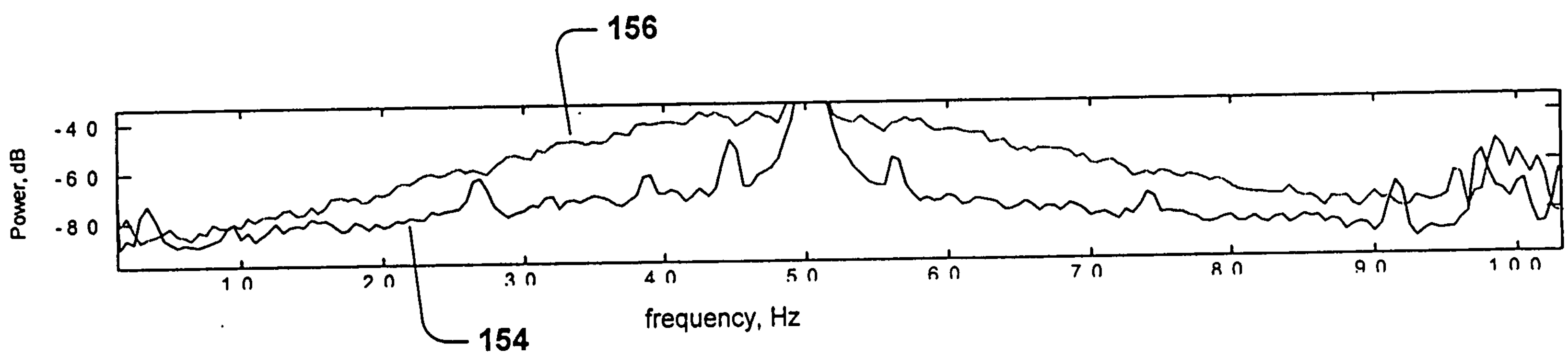

152
150
Power, dB
-50
-60
-70
-80
-90
10
20
30
40
50
60
70
80
90
100
frequency, Hz

Fig. 6b



156
154
Power, dB
-40
-60
-80
50
100
150
200
250
300
350
frequency, Hz

Fig. 6c



156
154
Power, dB
-40
-60
-80
10
20
30
40
50
60
70
80
90
100
frequency, Hz

Fig. 6d

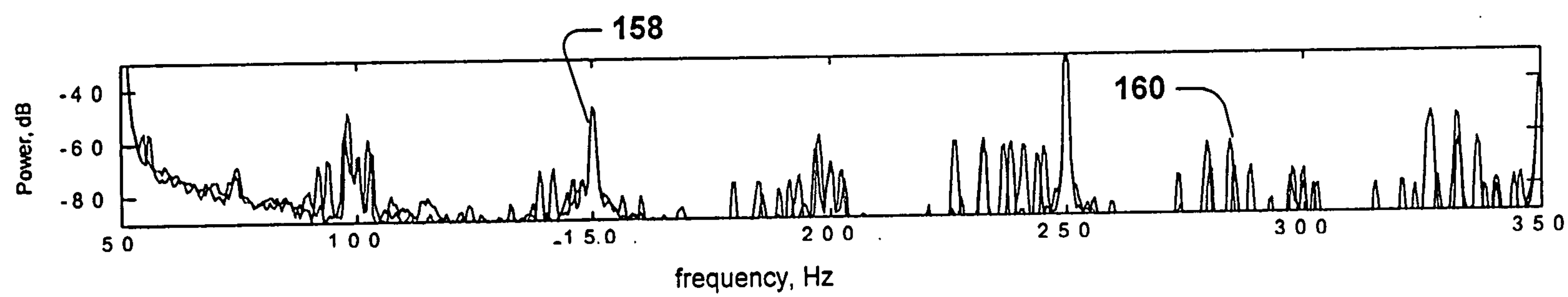

158
160
Power, dB
-40
-60
-80
50
100
150
200
250
300
350
frequency, Hz

Fig. 6e

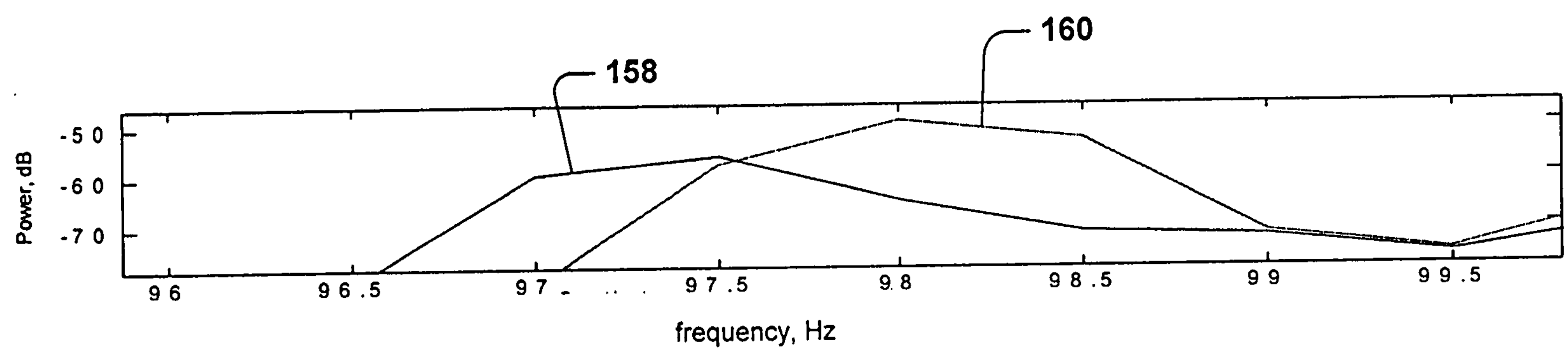

158
160
Power, dB
-50
-60
-70
96
96.5
97
97.5
98
98.5
99
99.5
frequency, Hz

Fig. 6f

200

202

f_0	f_1	f_2	f_3	f_4	•	•	•	f_n	
A_3	A_{34}	A_{67}	A_{78}	A_{84}	•	•	•	A_K	HEALTHY PUMP
A_{34}	A_{-68}	A_{-90}	A_{-65}	A_{45}	•	•	•	A_H	PUMP FAULT 1
A_{56}	A_{45}	A_{45}	A_{56}	A_{78}	•	•	•	A_X	PUMP FAULT 2
A_{-23}	A_{45}	A_7	A_{90}	A_{12}	•	•	•	A_Z	PUMP FAULT 3
A_{67}	A_{36}	A_3	A_{45}	A_{47}	•	•	•	A_X	PUMP FAULT 4
A_{78}	A_{87}	A_{12}	A_{67}	A_{37}	•	•	•	A_C	PUMP FAULT 5
A_{234}	A_{27}	A_{478}	A_{24}	A_{127}	•	•	•	A_Q	PUMP FAULT 6
A_{-98}	A_{78}	A_{26}	A_{12}	A_{128}	•	•	•	A_B	PUMP FAULT 7
A_{26}	A_{96}	A_{83}	A_{56}	A_{234}	•	•	•	A_M	PUMP FAULT 8
A_4	A_{32}	A_{187}	A_{56}	A_{34}	•	•	•	A_I	PUMP FAULT 9
A_0	A_{16}	A_{73}	A_{76}	A_{33}	•	•	•	A_E	PUMP FAULT N-1
A_{75}	A_{17}	A_{45}	A_{69}	A_{44}	•	•	•	A_Q	PUMP FAULT N

204

Fig. 6g

Divide the collected data into equal sets. Perform Hanning Windowing , FFT on each set to obtain 'Smoothed Periodogram' by averaging all the sets.

↓

Identify the fundamental supply component by locating the component having maximum amplitude in the stator current spectrum. Record its frequency (F_s) and amplitude (*FsAmp*). Locate multiples of F_s (supply related components)

↓

Calculate synchronous speed of the motor, $F_{sync} = F_s$ /polepairs. Locate the *slip frequency related* components by searching between ($mF_s - 2F_{slmin}$) and ($mF_s - 10F_{slmax}$) for m = 3,5 and 7.

$F_{slmax} = F_{sync}$ * maximum slip

$F_{slmin} = F_{sync}$ * minimum slip

↓

Calculate the *slip* from the above components. Locate $F_s + F_r$ and record its amplitude *FrAmp* where $F_r = F_{sync}$ * (1-*slip*)

↓

Search and locate the remaining 'slip frequency related' harmonics adjacent to other supply related components.

↓

Eliminate all the 'slip frequency related' harmonics between $F_s/2$ and $3F_s/2$ and measure the noise in the region.

noise_1 = [*sum of noise between* {(F_s - L -J) and (F_s - L)}+ {(F_s + L) and (F_s + L+J)}]

noise_i = [*sum of noise between* {(F_s - L -J(i+1)) and (F_s - L-Ji)}+ {(F_s + L+Ji) and (F_s + L+J(i+1))}]

for i = 2 to 5, L=6*resolution, and J= F_s /10

128

↓

Preprocess the attributes ***slip*,*FsAmp*, *SigAmp*, *Noise_1*, *Noise_2*, *Noise_3*, *Noise_4*** and ***Noise_5*** to make them acceptable by the Neural Network

130

Fig. 7

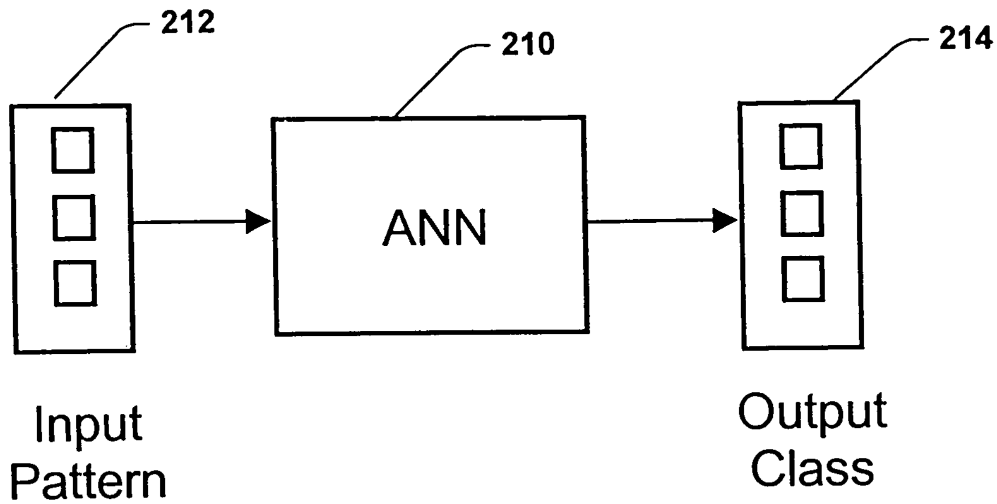

212
210
214
ANN
Input
Pattern
Output
Class

Fig. 8

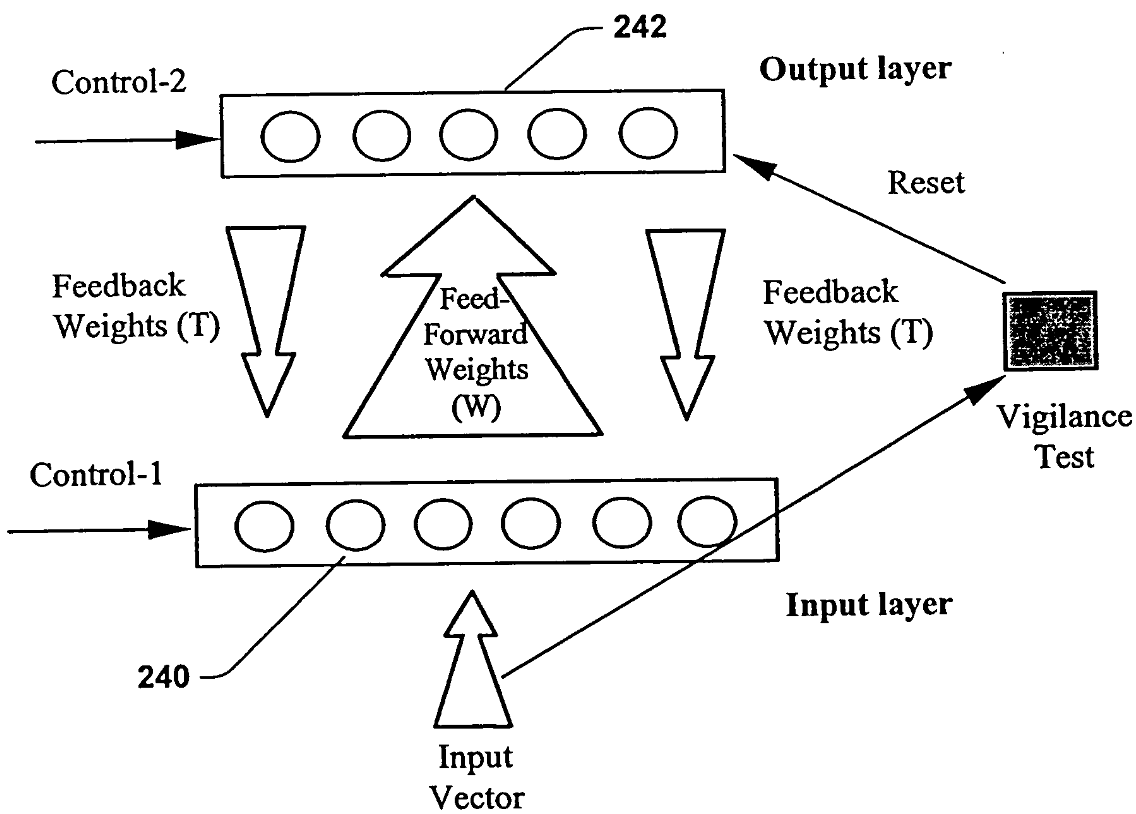

242
Control-2
Output layer
Reset
Feedback
Weights (T)
Feed-
Forward
Weights
(W)
Feedback
Weights (T)
Vigilance
Test
Control-1
Input layer
240
Input
Vector

Fig. 9

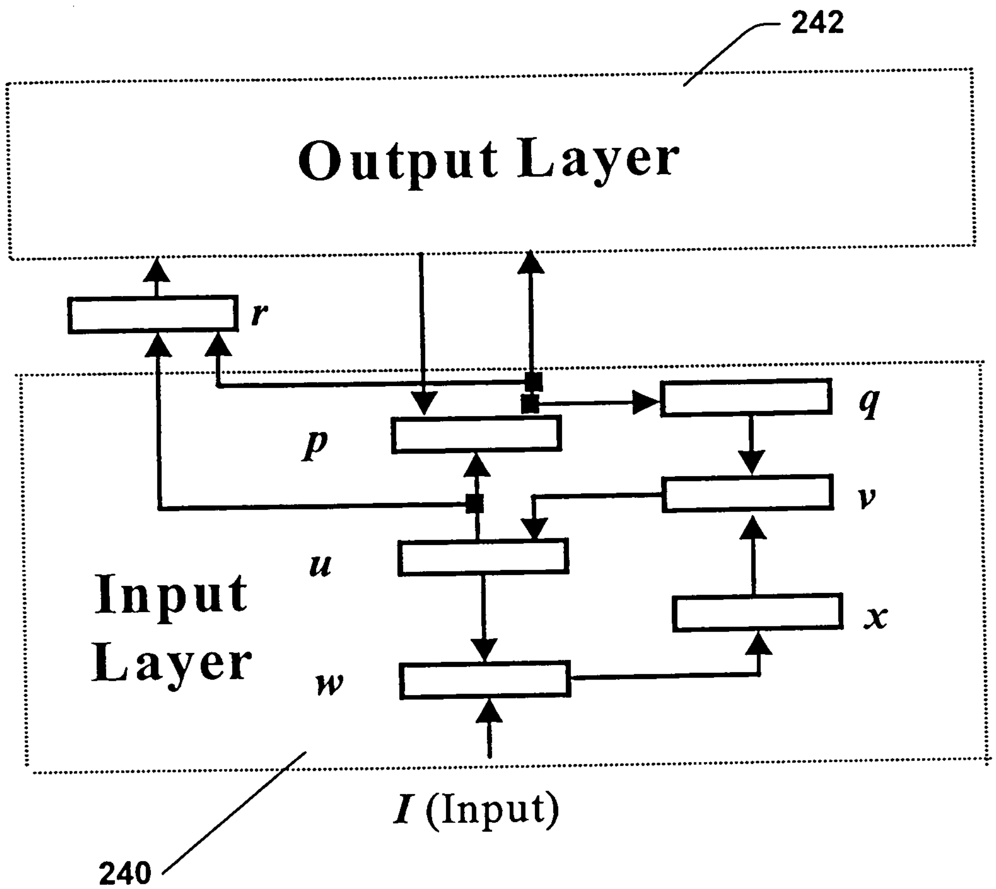
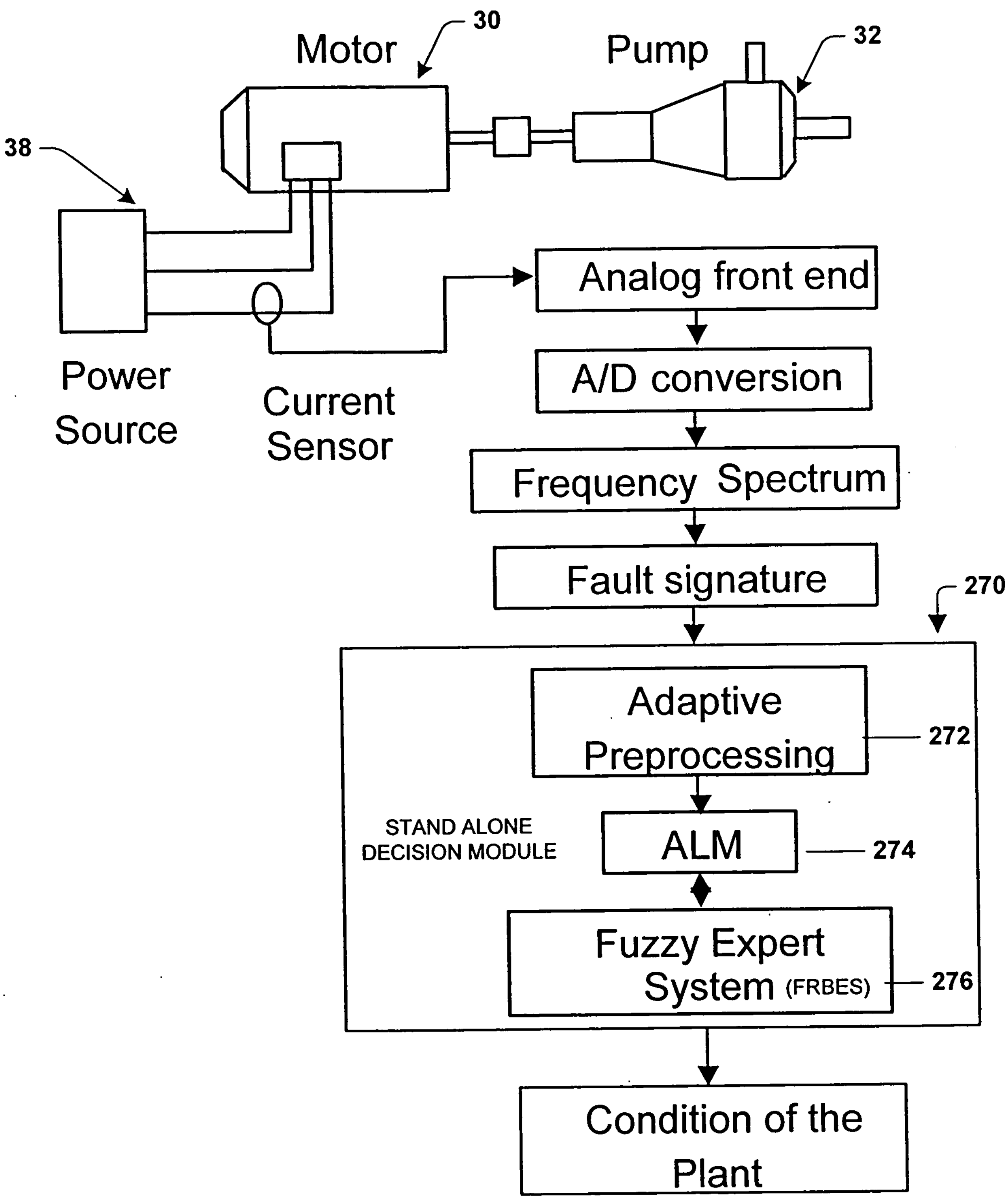
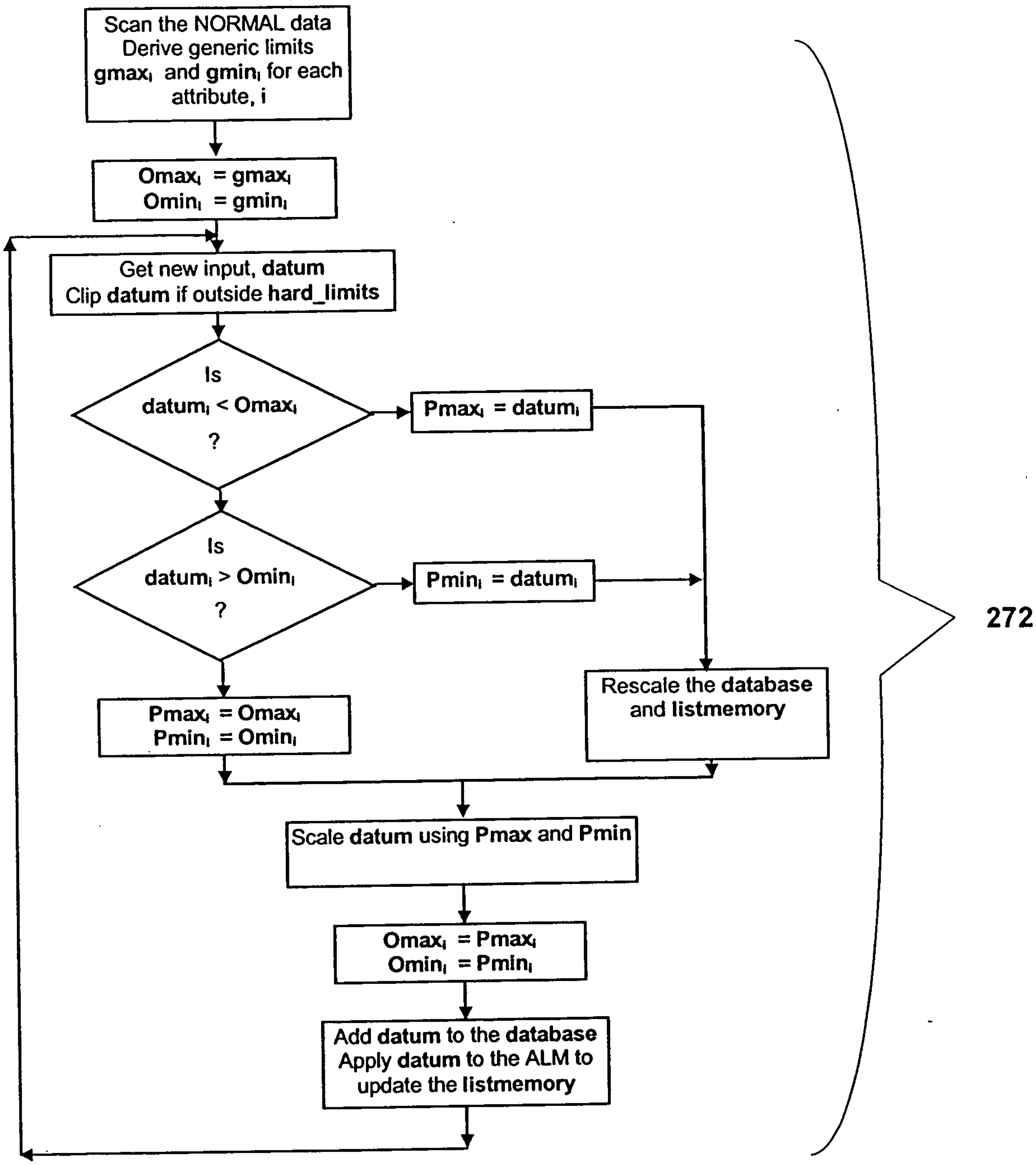


Fig. 10



30
Motor
Pump
32
38
Power Source
Current Sensor
Analog front end
A/D conversion
Frequency Spectrum
Fault signature
270
Adaptive Preprocessing
272
STAND ALONE DECISION MODULE
ALM
274
Fuzzy Expert System (FRBES)
276
Condition of the Plant

Fig. 11



Scan the NORMAL data
Derive generic limits
gmaxi and gmini for each attribute, i
Omaxi = gmaxi
Omini = gmini
Get new input, datum
Clip datum if outside hard_limits
Is
datumi < Omaxi
?
Pmaxi = datumi
Is
datumi > Omini
?
Pmini = datumi
Rescale the database and listmemory
Pmaxi = Omaxi
Pmini = Omini
Scale datum using Pmax and Pmin
Omaxi = Pmaxi
Omini = Pmini
Add datum to the database
Apply datum to the ALM to update the listmemory
272

Fig. 12

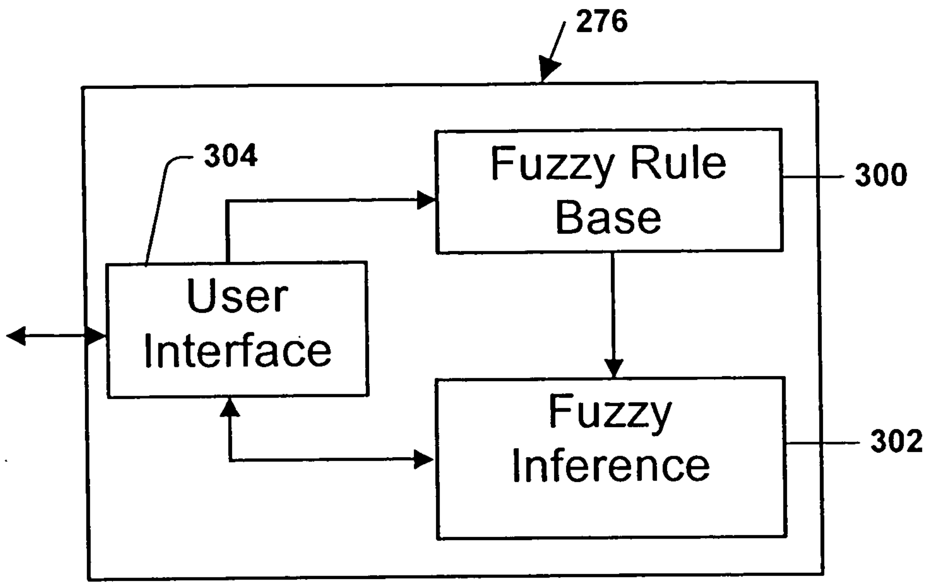

276
304
Fuzzy Rule Base
300
User Interface
Fuzzy Inference
302

Fig. 13

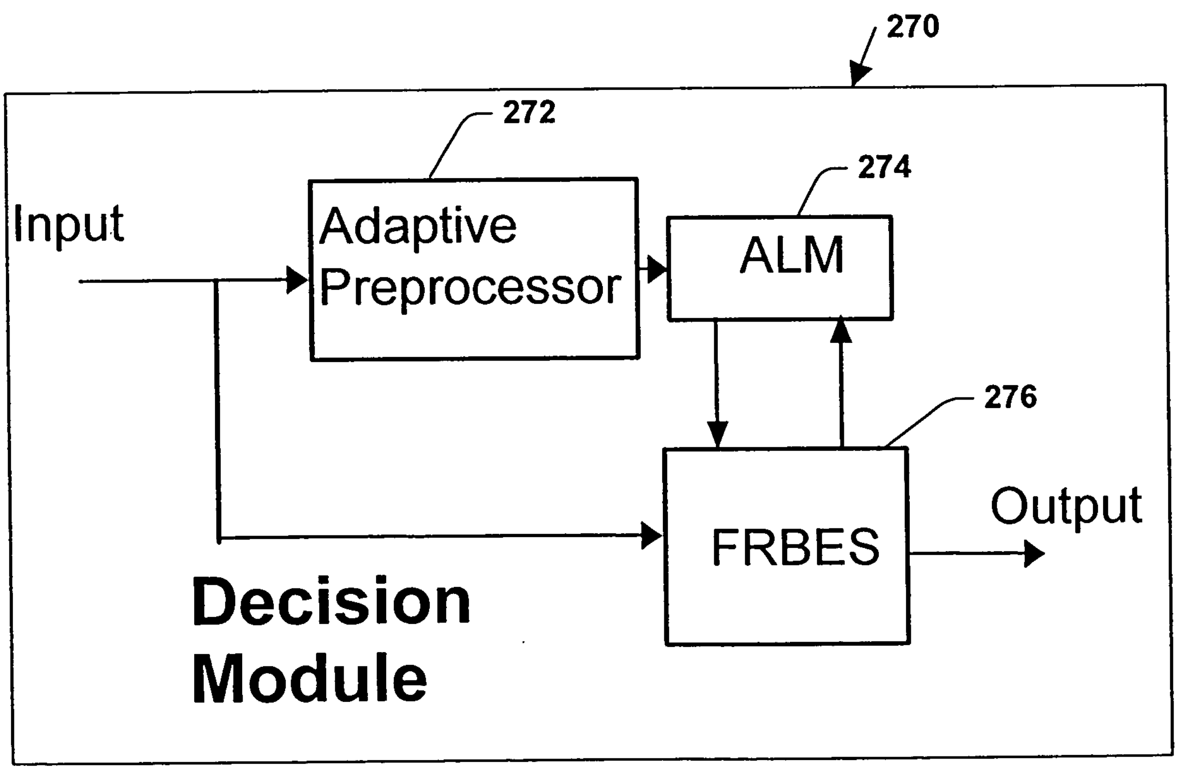

270
272
274
Input
Adaptive Preprocessor
ALM
276
FRBES
Output
Decision Module

Fig. 14

IF all the attributes are *NORMAL* THEN condition is normal
IF slip is *SLLO* and noise_2 is *H* THEN condition is low-cav
IF noise_4 and noise_5 are *VERH* THEN condition is sev-cav
IF noise_4 and noise_5 are *H* THEN condition is sev-cav
IF FsAmp is *SLLO* and noise_5 are *SLH* THEN condition is sev-cav
IF FsAmp is *LO* and noise_5 is *SLH* THEN condition is sev-cav
IF FsAmp is *VERLO* and noise_5 is *SLH* THEN condition is sev-cav
IF FsAmp is *SLLO* and noise_4 are *H* THEN condition is sev-cav
IF FsAmp is *LO* and noise_4 is *H* THEN condition is sev-cav
IF FsAmp is *LO* and noise_4 is *VERH* THEN condition is sev-cav
IF FsAmp is *SLLO* and slip is *SLLO* and noise_4 is *NORMAL* and noise_5 is *NORMAL* THEN condition is low-block
IF FsAmp is *LO* and noise_4 is *NORMAL* and noise_5 is *NORMAL* THEN condition is sev-block
IF slip and FsAmp are *VERLO* THEN condition is sev-block
IF FrAmp is *H* THEN condition is impel-fault
IF framp is *VERH* THEN condition is impel-fault

Fig. 15

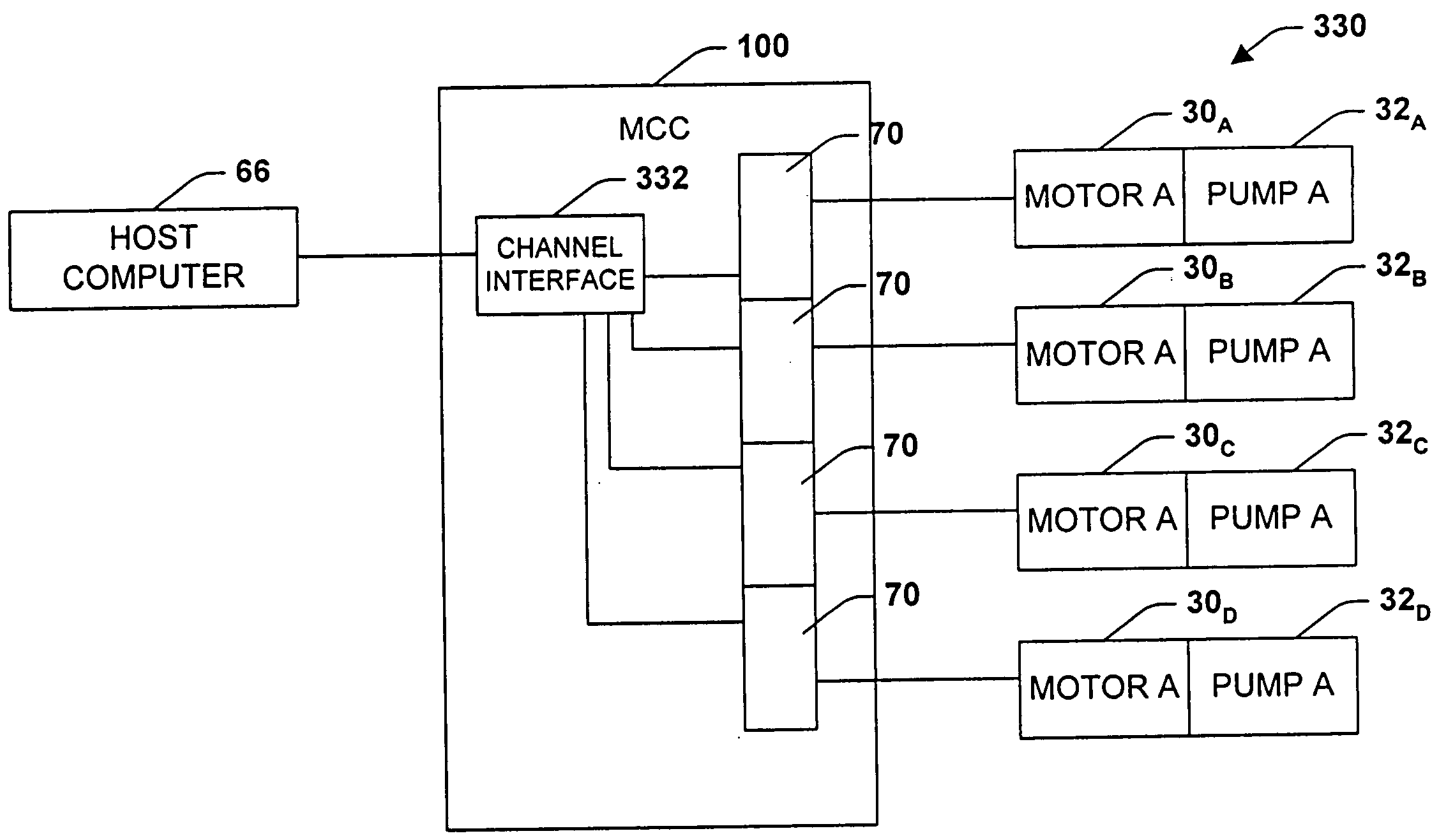

330
100
MCC
66
HOST COMPUTER
332
CHANNEL INTERFACE
70
70
70
70
30_A
32_A
MOTOR A
PUMP A
30_B
32_B
MOTOR A
PUMP A
30_C
32_C
MOTOR A
PUMP A
30_D
32_D
MOTOR A
PUMP A

Fig. 16

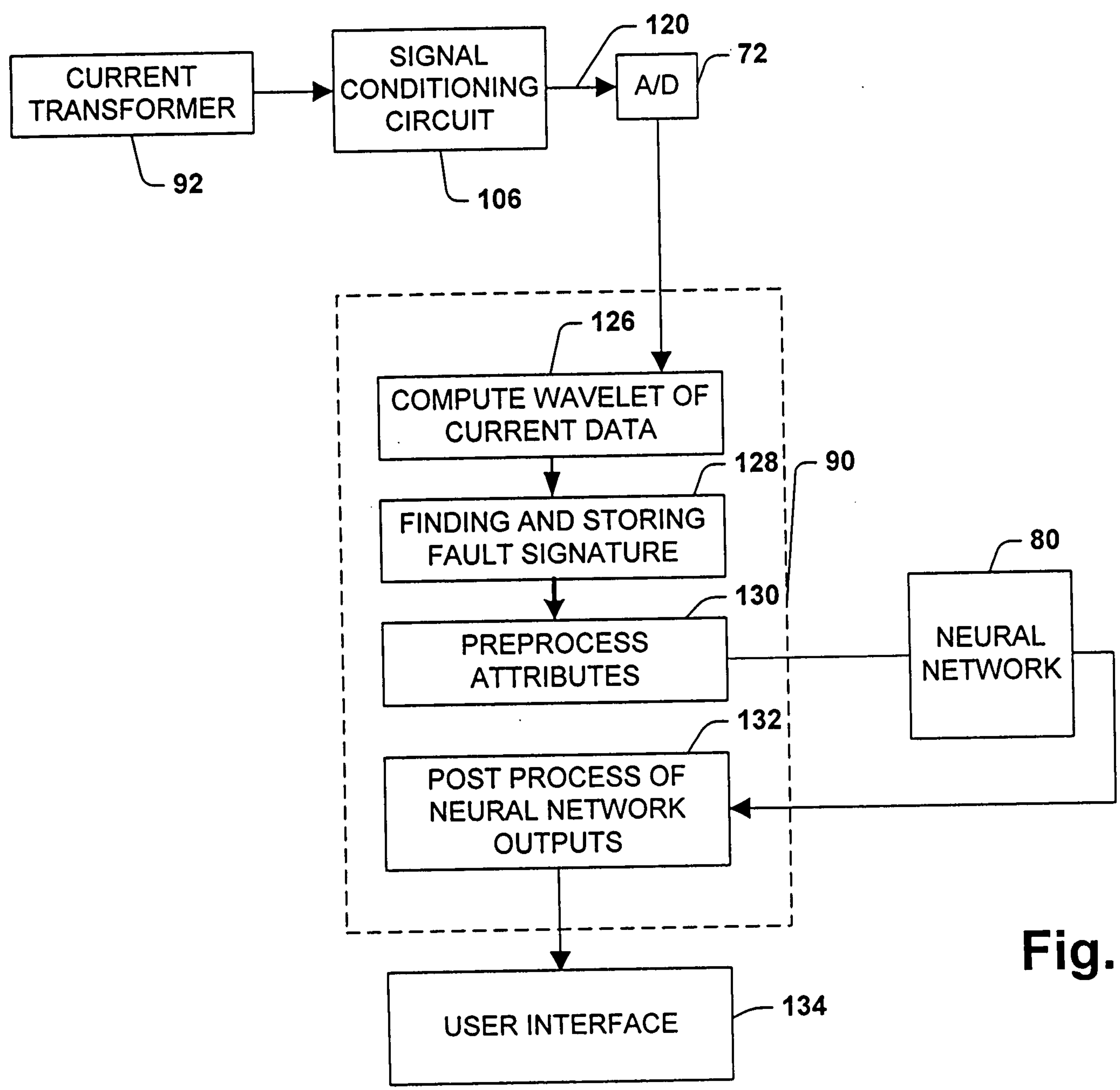

CURRENT TRANSFORMER
92
SIGNAL CONDITIONING CIRCUIT
106
120
A/D
72
126
COMPUTE WAVELET OF CURRENT DATA
128
FINDING AND STORING FAULT SIGNATURE
90
130
PREPROCESS ATTRIBUTES
80
NEURAL NETWORK
132
POST PROCESS OF NEURAL NETWORK OUTPUTS
USER INTERFACE
134

Fig. 17